

```
<?php
/* ----- */
/* */
/* v1.0 5.1.2017 Stephan Weber */
/* kml_gpx(kml, gpx) */
/* kml(kmz) in gpx umwandeln <trk> in gpx file schreiben */
/* optional: &wpt=n (oder irgendwas anderes hinterm '=': */
/* auch alle <wpt> ins gpx tun */
/* wie kann man daraus eine mediawiki extension machen? */
/*
http://www.zdnet.de/39151524/eigene-erweiterungen-fuer-mediawiki-schreiben/
*/
/* */
/* v1.1 3.5.2017 reine server-version */
/* */
/* aufruf
http://php.hutmachergass.de/kml_gpx.php?url=http://radreise-wiki.de/images/R
adweit_Münster_-_Bremen_(retour).kmz
*/
/* */
/* v1.2 8.5.2017 lösung: s.u. curl */
/* */
/* v1.3 10.5.2017 kommentare bereinigt */
/* gpx-files wurden nicht gelöscht, deswegen $gpxworkfile */
/* */
/* v1.4 11.5.2017 liste der downloads führen */
/* */
/* v1.5 12.5.2017 track name gefunden; sack und asche! */
/* */
/* v1.6 13.5.2017 rechenzeit ins log schreiben */
/* unlink($kmzfn); aus dem if-block raus, da gehörts nicht hin */
/* */
/* v1.7 15.5.2017 regex mit utf-8 chars */
/* sein gelassen, macht nicht viel sinn */
/* */
/* v1.8 19.5.2017 tempfilename unabhängig vom echten namen wählen */
/* logging der dauer des curl-aufrufs */
/* */
/* v1.9 20.5.2017 auf temp files ganz verzichten? */
/* zumindest auf gpxtmp */
/* logging mit ';' */
/* */
/* v2.0 9.6.2017 function hitliste() */
/* logfile lesen, hitliste machen und als html table ausgeben */
/* */
/* ----- */
/*
/*
kml to gpx converter, von hier, via stackoverflow:
http://thydzik.com/convert-from-google-kml-to-gps-exchange-format-gpx/
```

code-entwurf s. erste version dieses teils.

```
*/
/* ----- */

/* ----- */
/* ----- functions ----- */
/* ----- */

/* ----- */
/* helpers */
/* ----- */

function traceprint($tracebool, $printstring)
{
    if ($tracebool) echo $printstring;
} // traceprint()

/* ----- */
function traceprintr($tracebool, $arrname, $printarray)
{
    if ($tracebool)
    {
        echo "\ntraceprint $arrname\n";
        print_r($printarray);
    }
} // traceprintr()

/* ----- */
function microtime_float()
{
    list($usec, $sec) = explode(" ", microtime());
    return ((float)$usec + (float)$sec);
}

/* ----- */
function utcdatetime()
{
    return gmdate("Y-m-d\Th:i:s\Z");
}

/* ----- */
/* kml to gpx converter, von hier, via stackoverflow */
/* http://thydzik.com/convert-from-google-kml-to-gps-exchange-format-gpx/ */
/* ----- */
function kml_gpx($kml doc, &$gpx doc, &$ar0pts)
{
    if (null !== $kml doc)
    {
        $dom_kml = new DOMDocument();
        $dom_kml->loadXML($kml doc);
    }
}
```

```
$dom_gpx = new DOMDocument('1.0', 'UTF-8');
$dom_gpx->formatOutput = true;

//root node
$gpx = $dom_gpx->createElement('gpx');
$gpx = $dom_gpx->appendChild($gpx);

$gpx_version = $dom_gpx->createAttribute('version');
$gpx->appendChild($gpx_version);
$gpx_version_text = $dom_gpx->createTextNode('1.0');
$gpx_version->appendChild($gpx_version_text);

$gpx_creator = $dom_gpx->createAttribute('creator');
$gpx->appendChild($gpx_creator);
$gpx_creator_text = $dom_gpx->createTextNode('http://thydzik.com');
$gpx_creator->appendChild($gpx_creator_text);

$gpx_xmlns_xsi = $dom_gpx->createAttribute('xmlns:xsi');
$gpx->appendChild($gpx_xmlns_xsi);
$gpx_xmlns_xsi_text =
$dom_gpx->createTextNode('http://www.w3.org/2001/XMLSchema-instance');
$gpx_xmlns_xsi->appendChild($gpx_xmlns_xsi_text);

$gpx_xmlns = $dom_gpx->createAttribute('xmlns');
$gpx->appendChild($gpx_xmlns);
$gpx_xmlns_text =
$dom_gpx->createTextNode('http://www.topografix.com/GPX/1/0');
$gpx_xmlns->appendChild($gpx_xmlns_text);

$gpx_xsi_schemaLocation =
$dom_gpx->createAttribute('xsi:schemaLocation');
$gpx->appendChild($gpx_xsi_schemaLocation);
$gpx_xsi_schemaLocation_text =
$dom_gpx->createTextNode('http://www.topografix.com/GPX/1/0
http://www.topografix.com/GPX/1/0/gpx.xsd');
$gpx_xsi_schemaLocation->appendChild($gpx_xsi_schemaLocation_text);

$gpx_time = $dom_gpx->createElement('time');
$gpx_time = $gpx->appendChild($gpx_time);
$gpx_time_text = $dom_gpx->createTextNode(utcdate());
$gpx_time->appendChild($gpx_time_text);

// placemarks
$names = array();
foreach ($dom_kml->getElementsByTagName('Placemark') as $placemark)
{
    //name
    foreach ($placemark->getElementsByTagName('name') as $name)
    {
        $name = $name->nodeValue;
    }
}
```

```
//check if the key exists
if (array_key_exists($name, $names))
{
    //increment the value
    ++$names[$name];
    $name = $name." ({ $names[$name] })";
}
else
{
    $names[$name] = 0;
}

//description
$trkdesc = '';
foreach ($placemark->getElementsByTagName('description') as
$description)
{
    $description = $description->nodeValue;
    $trkdesc = $description;
}

// 12.5.2017 der track name steht in <Document><name>Saar</name> bzw
<Document><Placemark><name>Saar GPS-Track</name>
$trkname = '';
foreach ($placemark->getElementsByTagName('name') as $trackname)
{
    $trackname = $trackname->nodeValue;
    $trkname = $trackname;
}

if ($ar0pts['wpt'])
{
    foreach ($placemark->getElementsByTagName('Point') as $point)
    {
        foreach ($point->getElementsByTagName('coordinates') as
$coordinates)
        {
            //add the marker
            $coordinate = $coordinates->nodeValue;
            $coordinate = str_replace(" ", "", $coordinate); // trim
white space - sinnlos, wenn da \t und \n drin stehen!
            $coordinate = trim($coordinate);
            $latlng = explode(",", $coordinate);

            if (($lat = $latlng[1]) && ($lng = $latlng[0]))
            {
                $gpx_wpt = $dom_gpx->createElement('wpt');
                $gpx_wpt = $gpx->appendChild($gpx_wpt);

                $gpx_wpt_lat = $dom_gpx->createAttribute('lat');
```

```

$gpx_wpt->appendChild($gpx_wpt_lat);
$gpx_wpt_lat_text = $dom_gpx->createTextNode($lat);
$gpx_wpt_lat->appendChild($gpx_wpt_lat_text);

$gpx_wpt_lon = $dom_gpx->createAttribute('lon');
$gpx_wpt->appendChild($gpx_wpt_lon);
$gpx_wpt_lon_text = $dom_gpx->createTextNode($lng);
$gpx_wpt_lon->appendChild($gpx_wpt_lon_text);

// und wo bleibt die <ele>?? also:
if ($ele = $latlng[2])
{
    $gpx_ele = $dom_gpx->createElement('ele');
    $gpx_ele = $gpx_wpt->appendChild($gpx_ele);
    $gpx_ele_text = $dom_gpx->createTextNode($ele);
    $gpx_ele->appendChild($gpx_ele_text);
}

$gpx_name = $dom_gpx->createElement('name');
$gpx_name = $gpx_wpt->appendChild($gpx_name);
$gpx_name_text = $dom_gpx->createTextNode($name);
$gpx_name->appendChild($gpx_name_text);

$gpx_desc = $dom_gpx->createElement('desc');
$gpx_desc = $gpx_wpt->appendChild($gpx_desc);
$gpx_desc_text = $dom_gpx->createTextNode($description);
$gpx_desc->appendChild($gpx_desc_text);

$gpx_sym = $dom_gpx->createElement('sym');
$gpx_sym = $gpx_wpt->appendChild($gpx_sym);
$gpx_sym_text = $dom_gpx->createTextNode('Waypoint');
$gpx_sym->appendChild($gpx_sym_text);
}
}
} // if aopts wpt
}

foreach ($placemark->getElementsByTagName('LineString') as
$lineString)
{
    foreach ($lineString->getElementsByTagName('coordinates') as
$coordinates)
    {
        //add the new track
        $gpx_trk = $dom_gpx->createElement('trk');
        $gpx_trk = $gpx->appendChild($gpx_trk);

        $gpx_name = $dom_gpx->createElement('name');
        $gpx_name = $gpx_trk->appendChild($gpx_name);
        $gpx_name_text = $dom_gpx->createTextNode($trkname);
        // $gpx_name_text = $dom_gpx->createTextNode('track name');
    }
}

```

```
$gpx_name->appendChild($gpx_name_text);

// description aus der placemark description
$gpx_desc = $dom_gpx->createElement('desc');
$gpx_desc = $gpx_trk->appendChild($gpx_desc);
$gpx_desc_text = $dom_gpx->createTextNode($trkdesc);
$gpx_desc->appendChild($gpx_desc_text);

$gpx_trkseg = $dom_gpx->createElement('trkseg');
$gpx_trkseg = $gpx_trk->appendChild($gpx_trkseg);

$coordinates = $coordinates->nodeValue;
//split the coords by new line
$coordinates = preg_split("/[\s\r\n]+/", $coordinates);

foreach ($coordinates as $coordinate)
{
    if ('' != $coordinate)
        $latlng = explode(",", $coordinate);
    else {
        continue;
    }

    if (($lat = $latlng[1]) && ($lng = $latlng[0]))
    {
        $gpx_trkpt = $dom_gpx->createElement('trkpt');
        $gpx_trkpt = $gpx_trkseg->appendChild($gpx_trkpt);

        $gpx_trkpt_lat = $dom_gpx->createAttribute('lat');
        $gpx_trkpt->appendChild($gpx_trkpt_lat);
        $gpx_trkpt_lat_text = $dom_gpx->createTextNode($lat);
        $gpx_trkpt_lat->appendChild($gpx_trkpt_lat_text);

        $gpx_trkpt_lon = $dom_gpx->createAttribute('lon');
        $gpx_trkpt->appendChild($gpx_trkpt_lon);
        $gpx_trkpt_lon_text = $dom_gpx->createTextNode($lng);
        $gpx_trkpt_lon->appendChild($gpx_trkpt_lon_text);

        /* makes absolutely no sense to write the same time for
all trkpt

        $gpx_time = $dom_gpx->createElement('time');
        $gpx_time = $gpx_trkpt->appendChild($gpx_time);
        $gpx_time_text = $dom_gpx->createTextNode(utcdate());
        $gpx_time->appendChild($gpx_time_text);
        */

        // und wo bleibt die <ele>?? also:
        if ($ele = $latlng[2])
        {
            $gpx_ele = $dom_gpx->createElement('ele');
            $gpx_ele = $gpx_trkpt->appendChild($gpx_ele);
```

```

        $gpx_ele_text = $dom_gpx->createTextNode($ele);
        $gpx_ele->appendChild($gpx_ele_text);
    }
}
}
}
}
}
$gpxdoc = $dom_gpx->saveXML();
}

} // kml_gpx()

/* ----- */
/*
timestamp;computingtime;trackname
20170608230743;0.250;0stseeküstenradweg
*/
/* ----- */
function hitliste($logfn, &$arOpts)
{
if ((null !== $logfn) && ('' !== $logfn))
    $arlog = file($logfn, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
else die("no valid logfile");

$total = count($arlog);    // wird auch in der headline gebraucht

// überschrift
$arlogline = explode(';', $arlog[0]);
$datum = $arlogline[0];
$adatum = substr($datum, 6, 2).'.'.substr($datum, 4, 2).'.'.substr($datum,
0, 4);

$arlogline = explode(';', $arlog[$total - 1]);
$datum = $arlogline[0];
$edatum = substr($datum, 6, 2).'.'.substr($datum, 4, 2).'.'.substr($datum,
0, 4);

$arlogtracks = array();

foreach ($arlog as $logline)
{
list(, , $trackname) = explode(';', $logline);
$arlogtracks[] = $trackname;
} // foreach

unset($arlog); // no more needed

sort($arlogtracks, SORT_STRING); // array sortierter tracknamen

```

```
$artrackcount = array(); // array(trackname, anzahl)
$strkn = $arlogtracks[0]; // start value
$trkcnt = 0;
foreach ($arlogtracks as $trackname)
{
    if (0 == strcmp($strkn, $trackname))
        $trkcnt++;
    else
    {
        // anzahl + trackname in array schreiben
        $artrackcount[] = $trkcnt.';'.$strkn;
        $trkcnt = 1;
    }
    $strkn = $trackname;
} // foreach

unset($arlogtracks);

rsort($artrackcount, SORT_NUMERIC);

$htmltitle = 'Radreise-Wiki Rangliste Trackdownloads';
$headline = 'Rangliste von '.$adatum.' bis '.$edatum.' -
'.count($artrackcount).' Radrouten, '.$total.' GPX-Downloads';

// das folgende irgendwohin als template legen !!!
$htmlhead = '<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html><head><meta
charset="UTF-8"><title>'.$htmltitle.'</title></head>';

$htmlbody = '<body><h3>'.$headline.'</h3>';
$htmltable = '<table border cellpadding=0 cellspacing=0>
<tr><th>Anzahl</th><th>Track</th></tr>';
foreach ($artrackcount as $hitline)
{
    $items = explode(';', $hitline);
    $htmltable .=
"\n". '<tr><td>'.$items[0].</td><td>'.$items[1].</td></tr>';
}

$htmlbody .= $htmltable.</table></body></html>';

print($htmlhead.$htmlbody);

} // hitliste

/* ----- */
/* ----- */
/* --- main --- */
/* ----- */
/* ----- */
```



```
date_default_timezone_set('Europe/Paris');

$trace = 0; // wenn der höchste verwendete tracelevel 8 ist, dann
            // werden mit trace = 15 alle tracelevels angezeigt
            // 2: simple trace; 4: echo xml; 8: mysql; 128: input

$zeit = microtime_float();

if ($trace & 128)
{
    echo "<br>POST:";
    print_r($_POST);
    echo "<br>SERVER QUERYSTRING: ";
    echo $_SERVER['QUERY_STRING'];
    echo "<br>SERVER SERVERNAME: ";
    echo $_SERVER['SERVER_NAME'];
    exit();
}

$arOpts = array();
$arOpts['wpt'] = true;
$hitliste = false;

$kmzurl = '';

if ($_SERVER['QUERY_STRING'])
{
    $querystr = $_SERVER['QUERY_STRING'];

    if (" " != $querystr)
    {
        /* check for ?parms */
        parse_str($querystr);

        if ($url) $kmzurl = $url;

        if ($wpt) $arOpts['wpt'] = false;

        if ($hits) $hitliste = true;

        } /* args present */
    } // $_SERVER['QUERY_STRING']
else
    die("aufruf: ?url=<path-to-kmz>; option [&wpt=n] für gpx ohne
    waypoints");

/* ----- */
$workdir = './work/';
$logfn = $workdir.'log.txt';
```

```
if ($hitliste)
{
    hitliste($logfn, $arOpts);
    exit(0);
}

$urlparts = pathinfo($kmzurl);
$filename = $urlparts['filename'];
$kmztmpfn = $workdir.$azeit.'.kmz';    // code injection prevention

// create a new cURL resource
$ch = curl_init();

// set URL and other appropriate options
curl_setopt($ch, CURLOPT_URL, $kmzurl);
//return the transfer as a string
curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
curl_setopt($ch, CURLOPT_USERAGENT, 'Mozilla/5.0 (Windows; U; Windows NT
5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13');

// grab URL and pass it to $kmzcontents
$kmzcontents = curl_exec($ch);
$curlinfo = curl_getinfo($ch, CURLINFO_HTTP_CODE);
// close cURL resource, and free up system resources
curl_close($ch);

// zeit für den curl-aufruf loggen
$fcurlzeit = microtime_float() - $azeit;
$scurlzeit = sprintf("%2.3f", $fcurlzeit);

if (0 != strcmp ('200', $curlinfo))
    die("could not open $kmzurl ($scurlzeit s)");

// write stuff to local temp file
if (false === file_put_contents($kmztmpfn, $kmzcontents))
    die("write error with $kmztmpfn");

unset($kmzcontents);    // delete buffer

$kmldoc = '';
$zip = new ZipArchive;
if (true === $zip->open($kmztmpfn))
{
    // $zip->extractTo($dir);
    $fp = $zip->getStream('doc.kml');
    if (!$fp)
    {
        unlink($kmztmpfn);    // delete kmz file
        die("failed to get stream\n");
    }
}
```

```
while (!feof($fp))
    $kmldoc .= fread($fp, 2);

fclose($fp);
$zip->close();
}

unlink($kmztmpfn);    // delete kmz file

$gpxdoc = '';
kml_gpx($kmldoc, $gpxdoc, $arOpts);

header('Content-Disposition: attachment; filename='.$filename.'.gpx');
header('Content-type: text/xml');

print($gpxdoc);

$zeit = microtime_float();
$rechenzeit = sprintf("%2.3f", $zeit - $azzeit);
if (1.0 < $fcurlzeit) $rechenzeit = $rechenzeit."($scurlzeit)"; // curl
logging

// log
$logbuf = date("YmdHis").';'.$rechenzeit.';'.$filename."\n";
$logbuf = file_put_contents($logfn, $logbuf, FILE_APPEND);
unset($logbuf);

/* ----- */
?>
```

From:

<https://hutmachergass.de/dw-2016/> - **Stephans dokuwiki**

Permanent link:

<https://hutmachergass.de/dw-2016/doku.php?id=stephan:public:spielwiese>

Last update: **201710301014**